WHAT IS CLAIMED IS:

Sub a10

1. A central processing apparatus for assigning instructions of a program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units each executing the instructions by accessing a memory and a global register,

wherein the program comprises a plurality of instruction sequences, each instruction sequence comprises a plurality of instructions including a data dependency, a control dependency between the instruction sequences is represented by a commit instruction, and an instruction of data production or data consumption includes a flag representing possession of a register number in the global register,

the central processing apparatus, comprising:

a task window number generator configured to assign a task window number to the instruction sequences in a task window, the commit instruction being at the end of the task window;

an assignment unit configured to assign the instructions to the plurality of buffers, each instruction sequence corresponding to one buffer;

a register update unit configured to update data in the register number accessed by a particular instruction sequence in a task window if the particular instruction

Sub a10

sequence is accepted by the commit instruction in the task window; and

a memory update unit configured to update data in the memory address accessed by a particular instruction sequence in a task window if the particular instruction sequence is accepted by the commit instruction in the task window.

2. The central processing apparatus according to claim 1,

wherein the program comprises a plurality of task windows, each task window comprises a plurality of the instruction sequences, and each instruction sequence including non-speculative instructions is located before each instruction sequence including speculative instructions in the same task window.

3. The central processing apparatus according to claim 2,

wherein each instruction in the program is aligned in correspondence with each of the plurality of execution units to execute the each instruction in parallel, and a task number representing the instruction sequence and the execution unit is assigned to each instruction.

4. The central processing apparatus according to

Sub A10

~~claim 3,~~

~~wherein the commit instruction in the task window represents a condition instruction, and includes a condition, the task numbers to be accepted if the condition is not satisfied, and the task numbers to be rejected if the condition is satisfied.~~

~~5. The central processing apparatus according to claim 4,~~

~~wherein the commit instruction represents a branch condition instruction, and additionally includes a branch address of the instruction if the condition is satisfied.~~

~~6. The central processing apparatus according to claim 3,~~

~~wherein the commit instruction in the task window represents a loop condition instruction, and includes a loop condition, the task numbers representing a loop, and the task numbers to be accepted in case of the first loop only.~~

~~7. The central processing apparatus according to claim 3,~~

~~wherein the global register comprises a plurality of register numbers, and the flag of the instruction of data production or data consumption represents the register~~

Sub a10

~~number of non-use for other instructions until the instruction is completely executed.~~

~~8. The central processing apparatus according to claim 7,~~

~~further comprising an instruction decoder configured to decode a plurality of the instructions in order, and supply each instruction to said assignment unit along with operand data, the task number, and the task window number to which the instruction belongs.~~

~~9. The central processing apparatus according to claim 8, wherein,~~

~~when said instruction decoder decodes the instruction including the flag,~~

~~said instruction decoder sets the register number in the global register represented by the flag as non-use for the other instructions.~~

~~10. The central processing apparatus according to claim 3,~~

~~wherein each of the plurality of buffers comprises a plurality of queues, and each of the plurality of queues exclusively stores the instructions of predetermined task number by first in first out.~~

Sub a10

11. The central processing apparatus according to claim 10,

wherein said assignment unit assigns the instruction to the queue in the execution buffer corresponding to the task number of the instruction in the program.

12. The central processing apparatus according to claim 1,

wherein said task window number generator increments the task window number by one when the commit instruction is detected as the last instruction in the instruction sequence, and assigns an incremented task window number to the instruction sequences consisting of the instruction sequences from an instruction just behind the commit instruction to the next commit instruction in the program.

13. The central processing apparatus according to claim 9,

further comprising a plurality of local registers respectively connected to each of the plurality of execution units,

wherein each execution unit executes the instruction by accessing a respective local register in order to temporarily preserve the execution result of the instruction.

a10

14. the central processing apparatus according to claim 13, wherein,

if a particular instruction sequence is accepted by execution of the commit instruction in the task window,

said register update unit updates data in the register number of the global register represented by the flag in the particular instruction sequence using the execution result preserved in the local register.

15. The central processing apparatus according to claim 13, wherein,

if a particular instruction sequence is rejected by execution of the commit instruction in the task window,

said register update unit does not update data in the register number of the global register represented by the flag in the particular instruction sequence.

16. The central processing apparatus according to claim 1,

wherein said memory update unit temporarily preserves the execution result of a store instruction in a particular instruction sequence executed by the execution unit.

17. The central processing apparatus according to claim 16, wherein,

if the particular instruction sequence is accepted by

Sub a18

execution of the commit instruction in the task window,

said memory update unit updates data in the address of the memory represented by the store instruction using the preserved execution result.

18. The central processing apparatus according to claim 16, wherein,

if the particular instruction sequence is rejected by execution of the commit instruction in the task window,

said memory update unit does not update data in the address of the memory represented by the store instruction.

19. The central processing apparatus according to claim 16,

further comprising a load buffer to temporarily preserve a load instruction in order to read data from the memory or said memory update unit.

20. The central processing apparatus according to claim 19, wherein,

when the execution unit executes the load instruction in a particular instruction sequence accepted by the commit instruction,

the execution unit decides whether the load instruction depends on the execution result of the store instruction in said memory update unit.

Sub a10

21. The central processing apparatus according to claim 20, wherein,

if the load instruction depends on the execution result of the store instruction in said memory update unit,

the load buffer loads the execution result of the store instruction from said memory update unit.

22. The central processing apparatus according to claim 21, wherein,

if the load instruction does not depend on the execution result of the store instruction in said memory update unit,

load buffer loads data stored in the address of the memory represented by the load instruction.

23. A compile method for generating a program executed by a central processing apparatus for assigning instructions of the program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units executing the instruction by accessing a memory and a global register, comprising the steps of:

dividing the program into a plurality of instruction sequences each including a data dependency;

generating a commit instruction instead of a condition

Sub a10

instruction representing a control dependency between the instruction sequences in the program;

assigning a flag to an instruction of data production or data consumption, the flag representing possession of a register in the global register accessed by the instruction; and

assigning a task number to each instruction in the instruction sequences belonging to a task window, the commit instruction being at the end of the task window.

24. A computer readable memory containing computer readable instructions in a computer for assigning instructions of a program to a plurality of buffers each connected to one of a plurality of execution units, the plurality of execution units executing the instruction by accessing a memory and a global register, comprising:

an instruction means for causing the computer to divide the program into a plurality of instruction sequences each including a data dependency;

an instruction means for causing the computer to generate a commit instruction instead of a condition instruction representing a control dependency between the instruction sequences in the program;

an instruction means for causing the computer to assign a flag to an instruction of data production or data consumption, the flag representing possession of a register

Sub a10

~~in the global register accessed by the instruction; and~~

~~an instruction means for causing the computer to assign a task number to each instruction in the instruction sequences belonging to a task window, the commit instruction being at the end of the task window.~~